

CLIPPEDIMAGE= EP000565194A1

PUB-NO: EP000565194A1

DOCUMENT-IDENTIFIER: EP 565194 A1

TITLE: Method to store and retrieve return addresses.

PUBN-DATE: October 13, 1993

INVENTOR-INFORMATION:

NAME	COUNTRY
BAUER, HARALD	DE
HELLWIG, KARL	DE
LORENZ, DIETMAR	DE
SCHUCK, JOHANNES DR	DE

ASSIGNEE-INFORMATION:

NAME	COUNTRY
PHILIPS PATENTVERWALTUNG	DE
PHILIPS ELECTRONICS NV	NL

APPL-NO: EP93201005

APPL-DATE: April 6, 1993

PRIORITY-DATA: DE04211966A (April 9, 1992)

INT-CL_(IPC): G06F009/42

EUR-CL (EPC): G06F009/42

ABSTRACT:

To store return addresses, a special stack comprising registers is used, particularly in rapid signal processors, which can be written to or read out from, in particular within a clock cycle. The number of interleaved programs is, however, then limited by the size of the stack. If the data memory or an area in it is intended to be used as a stack to avoid a limitation of this type, the storage and, in particular, the reading out of return addresses in each case requires memory access which cannot be gained within a clock cycle. Therefore, according to the invention, an additional register is used into

which the return addresses can be written in one program branch and, for example, in parallel with the execution of the following command, the contents of this register are transferred to the stack area of the data memory. A following return can then be quickly carried out with the register data. So that a plurality of consecutive returns can also be carried out in this fashion, the return addresses of the preceding program branch are transferred after each return from the data memory to the register, likewise in parallel with the execution of the next command. Program branches and returns can thus be carried out quickly without the command process being hindered or delayed through the accessing of the stack in the data memory. <IMAGE>

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) Veröffentlichungsnummer: **0 565 194 A1**

(12)

EUROPÄISCHE PATENTANMELDUNG

(21) Anmeldenummer: 93201005.1

(51) Int. Cl.⁵: G06F 9/42

(22) Anmeldetag: 06.04.93

(30) Priorität: 09.04.92 DE 4211966

(43) Veröffentlichungstag der Anmeldung:
13.10.93 Patentblatt 93/41

(84) Benannte Vertragsstaaten:
DE FR GB

(71) Anmelder: Philips Patentverwaltung GmbH
Wendenstrasse 35c
D-20097 Hamburg(DE)

(84) DE

(71) Anmelder: PHILIPS ELECTRONICS N.V.
Groenewoudseweg 1
NL-5621 BA Eindhoven(NL)

(84) FR GB

(72) Erfinder: Bauer, Harald, c/o Philips
Patentverwaltung GmbH

Wendenstrasse 35

W-2000 Hamburg 1(DE)

Erfinder: Hellwig, Karl, c/o Philips

Patentverwaltung GmbH

Wendenstrasse 35

W-2000 Hamburg 1(DE)

Erfinder: Lorenz, Dietmar, c/o Philips

Patentverwaltung GmbH

Wendenstrasse 35

W-2000 Hamburg 1(DE)

Erfinder: Schuck, Johannes, Dr., c/o Philips

Patentver. GmbH

Wendenstrasse 35

W-2000 Hamburg 1(DE)

(74) Vertreter: Peters, Carl Heinrich, Dipl.-Ing. et al
Philips Patentverwaltung GmbH,
Wendenstrasse 35c
D-20097 Hamburg (DE)

(54) Anordnung zum Abspeichern und Auslesen von Rücksprungsdaten.

(57) Für die Speicherung von Rücksprungsdaten wird insbesondere bei schnellen Signalprozessoren ein besonderer, aus Registern aufgebauter Stack benutzt, der insbesondere innerhalb eines Taktzyklus eingeschrieben bzw. ausgelesen werden kann. Die Anzahl der verschachtelten Programme ist dann allerdings durch die Größe des Stacks begrenzt. Wenn zur Vermeidung einer solchen Begrenzung der Datenspeicher bzw. ein Bereich darin als Stack benutzt werden soll, erfordert das Abspeichern und insbesondere das Auslesen von Rücksprungsdaten jeweils einen Speicherzugriff, der nicht innerhalb eines Taktzyklus abgewickelt werden kann. Erfindungsgemäß wird daher ein zusätzliches Register verwendet, in das bei einem Programmsprung die Rücksprungsdaten eingeschrieben werden, und bei-

spielsweise parallel zur Ausführung des folgenden Befehls wird der Inhalt dieses Registers in den Stackbereich des Datenspeichers übertragen. Ein folgender Rücksprung kann dann schnell mit den Daten des Registers durchgeführt werden. Damit auch mehrere aufeinanderfolgende Rücksprünge auf diese Weise durchgeführt werden können, werden nach jedem Rücksprung die Rücksprungsdaten des vorhergehenden Programmsprungs aus dem Datenspeicher in das Register übertragen, und zwar ebenfalls parallel zur Ausführung des nächsten Befehls. Dadurch können Programmsprünge und Rücksprünge schnell ausgeführt werden, ohne daß der Befehlsablauf durch Zugriffe auf den Stack im Datenspeicher behindert bzw. verzögert wird.

EP 0 565 194 A1

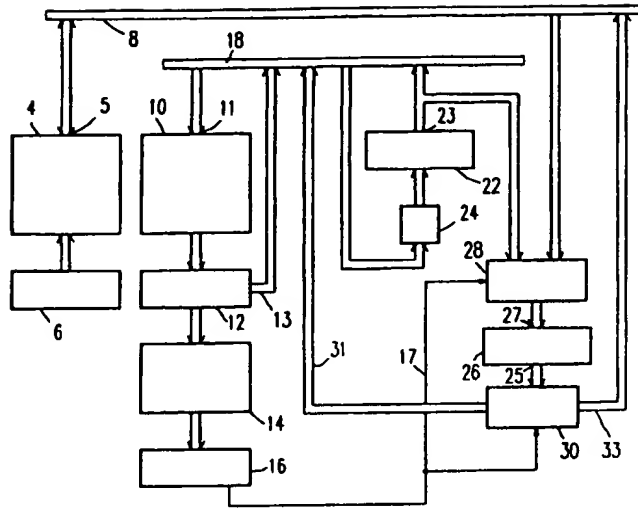


Fig.2

Die Erfindung betrifft eine Anordnung zum Abspeichern und Auslesen von Rücksprungsdaten, insbesondere Programmadressen, bei einem Programmsprung und einem Rücksprung in einer Datenverarbeitungsanlage, die einen Datenspeicher großer Kapazität mit wenigstens einem Datenanschluß, einen Programmspeicher mit einem Programmadreßgenerator und eine Speicheranordnung für die Rücksprungsdaten aufweist, wobei die Speicheranordnung bei einem Programmsprung die Rücksprungsdaten aufnimmt und bei einem Rücksprung die zuletzt aufgenommenen Rücksprungsdaten abgibt.

Bei einem Programmsprung in einer Datenverarbeitungsanlage, der beispielsweise durch den Aufruf eines Unterprogramms oder durch ein externes Unterbrechungssignal ausgelöst wird, ist es üblich, die zuletzt erzeugte Adresse des Programmadreßgenerators, der allgemein auch als Programmzähler oder Befehlszähler bezeichnet wird, abzuspeichern, damit nach Beendigung des Unterprogramms das übergeordnete Programm unmittelbar an der unterbrochenen Stelle fortgesetzt werden kann. Dabei kann der Fall auftreten, daß ein Unterprogramm, das durch einen Programmsprung oder ein Unterbrechungssignal aufgerufen wurde, während des Ablaufs seinerseits unterbrochen wird, so daß zeitweise eine mehrfache Verschachtelung von Unterprogrammen vorliegen kann. Die Rücksprungsdaten von allen unterbrochenen Unterprogrammen oder Programmteilen werden in der Reihenfolge ihres Auftretens gespeichert, wobei die zuletzt gespeicherten Daten als erstes wieder ausgelesen werden. Hierfür dient insbesondere bei schnellen Signalprozessoren eine besondere Speicheranordnung, die auch mit Stack bezeichnet wird und aus einer Registerbank besteht. Das Abspeichern und das Aufrufen von Rücksprungsdaten erfolgt dabei sehr schnell, nämlich innerhalb eines Taktzyklus. Die Anzahl von Unterprogrammen, die ineinander verschachtelt sein können, ist dabei aber durch die Größe der für den Stack verwendeten Registerbank begrenzt.

Aus diesem Grunde wird bei allgemeinen Mikroprozessoren, die also nicht speziell für die digitale Signalverarbeitung ausgelegt sind, häufig als Stack ein Teil des allgemeinen Datenspeichers benutzt. Bei sehr zeitkritischen Verarbeitungen, wie dies bei der digitalen Signalverarbeitung der Fall ist, ist dabei jedoch nachteilig, daß das Abspeichern der Rücksprungsdaten und noch mehr das Aufrufen der Rücksprungsdaten eine erhebliche Zeit in Anspruch nimmt, die insbesondere von der Geschwindigkeit des Datenspeichers großer Kapazität abhängt. Andererseits ist bei Verwendung des Datenspeichers als Stack praktisch keine Begrenzung der Anzahl ineinander verschachtelter Unterprogramme gegeben.

Aufgabe der Erfindung ist es, eine Anordnung der eingangs genannten Art anzugeben, bei der der Umfang des Stacks praktisch nicht begrenzt ist, wobei jedoch trotzdem eine schnelle Abspeicherung und insbesondere ein schneller Aufruf der Rücksprungsdaten möglich ist.

Diese Aufgabe wird erfindungsgemäß dadurch gelöst, daß bei Verwendung des Datenspeichers als Speicheranordnung ein zusätzliches Register vorgesehen ist, dessen Ausgang bei einem Rücksprung mit dem Adresseneingang des Programmspeichers und nach einem Programmsprung mit dem Datenanschluß des Datenspeichers koppelbar ist und dessen Eingang bei einem Programmsprung mit dem Adressenausgang des Programmadreßgenerators und nach einem Programmrücksprung mit dem Datenanschluß des Datenspeichers koppelbar ist.

Auf diese Weise werden die Vorteile des nahezu unbegrenzten Stacks im Datenspeicher mit der Geschwindigkeit einer gesonderten Speicheranordnung aus Registern für das Stack verbunden. Falls ein Programmsprung auftritt, werden die Rücksprungsdaten ohne Zeitverzögerung, d.h. innerhalb eines Taktzyklus, im zusätzlichen Register gespeichert, während der Adreßsprung ausgeführt wird. Nach dem Abspeichern der Rücksprungsdaten im zusätzlichen Register wird dessen Inhalt in den Datenspeicher an die Stelle des nächsten freien Speicherplatzes im Stack übertragen, während aus dem Programmspeicher bereits der nächste Befehl aufgerufen und verarbeitet wird. Das zusätzliche Register behält dabei seinen Inhalt. Dieses Abspeichern der Rücksprungsdaten aus dem zusätzlichen Register kann aber auch später erfolgen, spätestens dann, wenn ein weiterer Programmsprung auftritt, wobei dann die neuen Rücksprungsdaten in das zusätzliche Register übertragen werden und quasi gleichzeitig dessen alter Inhalt in den Datenspeicher an den nächsten freien Speicherplatz im Stack eingeschrieben wird. Bei einem auftretenden Rücksprung werden die zuletzt abgespeicherten Rücksprungsdaten unmittelbar ohne Zeitverzögerung ausgelesen, indem der Ausgang des zusätzlichen Registers mit dem Adreßeingang des Programmspeichers verbunden wird, und anschließend werden die vorhergehend abgespeicherten Rücksprungsdaten aus dem Datenspeicher ausgelesen und in das zusätzliche Register geladen, während aus dem Programmspeicher bereits der erste Befehl der Rücksprungstelle ausgelesen und verarbeitet wird. In beiden Fällen erfolgt die Verarbeitung der Rücksprungsdaten innerhalb eines einzigen Taktzyklus, während das eigentliche Abspeichern bzw. Auslesen der Rücksprungsdaten aus dem Datenspeicher quasi im Hintergrund parallel zur jeweils nächsten Befehlsausführung erfolgt.

Ein Ausführungsbeispiel der Erfindung wird nachstehend anhand der Zeichnung erläutert.

Es zeigen

Fig. 1 eine bekannte Anordnung mit einer gesonderten Speicheranordnung für den Stack,

Fig. 2 eine Anordnung nach der Erfindung mit einem zusätzlichen Register.

Die Fig. 1 zeigt einen Programmspeicher 10, der die auszuführenden Befehle des Programms bzw. der einzelnen Programme enthält. Jedes ausgelesene Befehlswort wird in ein Instruktionsregister 12 geladen, dessen Ausgang mit einem Befehlsdecoder 14 verbunden ist. Die decodierten Steuersignale werden in einem Befehlsausgangsregister 16 gespeichert, wo sie zur Verfügung stehen, während der nächste Befehl bereits entschlüsselt wird.

Die Adressen des Programmspeichers 10 werden über einen Adressenbus 18 zugeführt, der mit dem Ausgang 23 eines Programmzählregisters 22 verbunden ist. Bei einem Programmablauf ohne Unterbrechungen bzw. Aufruf von Unterprogrammen wird nach jedem aus dem Programmspeicher 10 ausgelesenen Befehl das Programmzählregister um eine Adresse weitergeschaltet. Dies erfolgt durch einen Addierer 24, der die vom Programmzählregister 22 abgegebene Adresse um 1 erhöht dem Eingang des Programmzählregisters 22 zuführt.

Bei einem Programmsprung, der stets durch einen Befehl erfolgt, wird die im Befehl enthaltene neue Adresse über die Verbindung 13 dem Adreßbus 18 zugeführt. Gleichzeitig wird durch weitere, im Decoder 14 entschlüsselte Teile des Befehls über das Ausgangsregister 16 und die Leitung 17 ein Speicher 2 angesteuert, der die Rücksprungdaten aufnimmt und auch als Stack bezeichnet wird. Die Rücksprungdaten sind insbesondere die Adresse des Programmzählregisters 22, die in diesem Moment am Ausgang 23 abgegeben werden. Diese werden über den Eingang 1 des Stacks zugeführt und im nächsten freien Register darin abgespeichert. Gleichzeitig wird auch die Sprungadresse, die über die Verbindung 13 und den Adreßbus 18 dem Adresseneingang des Programmspeichers 10 zugeführt wird, über den Addierer 24 um 1 erhöht dem Programmzählregister 22 zugeführt. Damit werden dann die weiteren Befehle des Unterprogramms aufgerufen.

Wenn ein Unterprogramm endet, wird durch einen entsprechenden Befehl an dessen Ende auf der Verbindung 17 ein Signal erzeugt, das die zuletzt eingeschriebenen Rücksprungdaten aus dem Stack 2 ausliest und am Ausgang abgibt, der nun mit dem Adreßbus 18 gekoppelt ist. Damit springt die dem Programmspeicher 10 zugeführte Adresse nun an die Stelle zurück, wo der vorhergehende Programmsprung erfolgt ist. Da der Stack 2

insbesondere aus Registern aufgebaut ist, können die Informationen darin schnell eingeschrieben und daraus ausgelesen werden, nämlich innerhalb eines einzigen Taktzyklus.

Wenn nun anstelle des besonderen, aus Registern aufgebauten Stacks 2 ein Bereich des Datenspeichers für den Stack verwendet werden soll, muß der Datenausgang 23 des Programmzählregisters 22 mit dem Datenanschluß des Datenspeichers verbunden werden, und entsprechend muß der Datenausgang bzw. Datenanschluß des Datenspeichers für das Abgeben der Rücksprungdaten mit dem Adreßbus 18 koppelbar sein. Dann entstehen jedoch sowohl beim Einschreiben von Rücksprungdaten infolge eines Programmsprungs oder beim Auslesen der Rücksprungdaten infolge eines Rücksprungs erhebliche Wartezeiten, da ein Speicherzyklus des Datenspeichers, der eine große Kapazität aufweist, allgemein länger dauert als eine Taktphase eines schnell arbeitenden Signalprozessors.

Bei der Anordnung nach Fig. 2 ist der Datenanschluß 5 eines Datenspeichers 4 nun nicht direkt mit dem Ausgang 23 des Programmzählregisters 22 verbunden, sondern über ein zusätzliches Register 26, dessen Eingang 27 ein Multiplexer 28 vorgeschaltet und dessen Ausgang 25 ein Demultiplexer 30 nachgeschaltet ist.

Während des Programmablaufs ohne Unterbrechung ist die Funktion dieser Anordnung gleich wie die Anordnung nach Fig. 1. Bei einem Programmsprung wird wieder die Adresse des Anfangs des Unterprogramms über die Verbindung 13 dem Adreßbus 18 und damit dem Adresseneingang 11 des Programmspeichers 10 zugeführt. Gleichzeitig wird durch ein entsprechendes Signal auf der Verbindung 17 der Eingang 27 des Registers 26 über den Multiplexer 28 mit dem Ausgang 23 des Programmzählregisters 22 verbunden und dessen Adresse in das Register 26 eingeschrieben. Dies kann ohne Zeitverlust, d.h. innerhalb eines Taktzyklus, vorgenommen werden. Während aus dem Programmspeicher 10 nun die erste Adresse des Unterprogramms ausgelesen und verarbeitet wird, verbindet der Demultiplexer 30 den Ausgang 25 des Registers 26, das nun insbesondere die Rücksprungadresse enthält, mit der Verbindung 33, so daß diese Rücksprungadresse nun über den Datenbus 8 dem Datenanschluß 5 des Datenspeichers 4 zugeführt wird und mittels entsprechender Adreßsteuerung aus dem Adreßgenerator 6 an der nächsten freien Stelle des Stackbereichs eingeschrieben wird.

Wenn nun ein weiterer Programmsprung auftritt, wird der vorhergehende Inhalt des Registers 26 mit den neuen Rücksprungdaten überschrieben, die ebenfalls parallel zur Befehlsausführung des nun angesprungenen Unterprogramms in den Da-

tenspeicher 4 eingeschrieben wird. Das Register 26 enthält somit stets nur die letzten Rücksprungdaten, während nur der Datenspeicher 4 auch die vorhergehenden Rücksprungdaten enthält.

Bei einem Rücksprung wird nun über ein entsprechendes Signal auf der Verbindung 17 der Demultiplexer 30 so umgeschaltet, daß der Ausgang 25 des Registers 26 über die Verbindung 31 mit dem Adreßbus 18 und damit mit dem Adreßeingang 11 des Programmspeichers 10 verbunden wird. Die Rücksprungdaten stehen also unmittelbar zur Verfügung, ohne daß dafür ein Speicherzugriff auf dem Datenspeicher 4 erforderlich ist.

Wenn nun jedoch ein weiterer Rücksprung auftritt, ohne daß vorher ein Programmsprung erfolgt ist, würde das Register 26 nun nicht die für den weiteren Rücksprung erforderlichen Rücksprungdaten enthalten. Daher wird unmittelbar nach einem Rücksprung der Multiplexer 28 umgeschaltet, so daß der Eingang 27 des Registers 26 über die Verbindung 35 und den Datenbus mit dem Datenanschluß 5 des Datenspeichers 4 verbunden ist, und durch entsprechende Adressierung durch den Adressengenerator 6 werden aus dem Stackbereich des Datenspeichers 4 die Rücksprungdaten des vorhergehenden Programmsprungs ausgelesen und in das Register 26 übertragen. Dies kann erfolgen, während der erste Befehl nach dem Rücksprung bereits aus dem Programmspeicher 10 ausgelesen und verarbeitet wird. Somit enthält das Register 26 stets die für den nächsten Rücksprung erforderlichen Rücksprungdaten.

Durch das zusätzliche Register 26 ist es also möglich, Programmsprünge und auch Rücksprünge ohne Zeitverlust durchzuführen und dennoch den Datenspeicher 4 für den Stack zu benutzen, so daß nahezu beliebig viele verschachtelte Unterprogramme auftreten können.

Patentansprüche

1. Anordnung zum Abspeichern und Auslesen von Rücksprungdaten, insbesondere Programmadressen, bei einem Programmsprung und einem Rücksprung in einer Datenverarbeitungsanlage, die einen Datenspeicher großer Kapazität mit wenigstens einem Datenanschluß, einen Programmspeicher mit einem Programmadreßgenerator und eine Speicheranordnung für die Rücksprungdaten aufweist, wobei die Speicheranordnung bei einem Programmsprung die Rücksprungdaten aufnimmt und bei einem Rücksprung die zuletzt aufgenommenen Rücksprungdaten abgibt, dadurch gekennzeichnet, daß bei Verwendung des Datenspeichers (4) als Speicheranordnung (2) ein zusätzliches Register (26) vorgesehen ist, dessen Ausgang (25) bei einem Rücksprung mit dem Adresseneingang (11) des Programmspeichers (10)

und nach einem Programmsprung mit dem Datenanschluß (5) des Datenspeichers (4) koppelbar ist und dessen Eingang (27) bei einem Programmsprung mit dem Adressenausgang (23) des Programmadreßgenerators (22) und nach einem Programmrückprung mit dem Datenanschluß (5) des Datenspeichers (4) koppelbar ist.

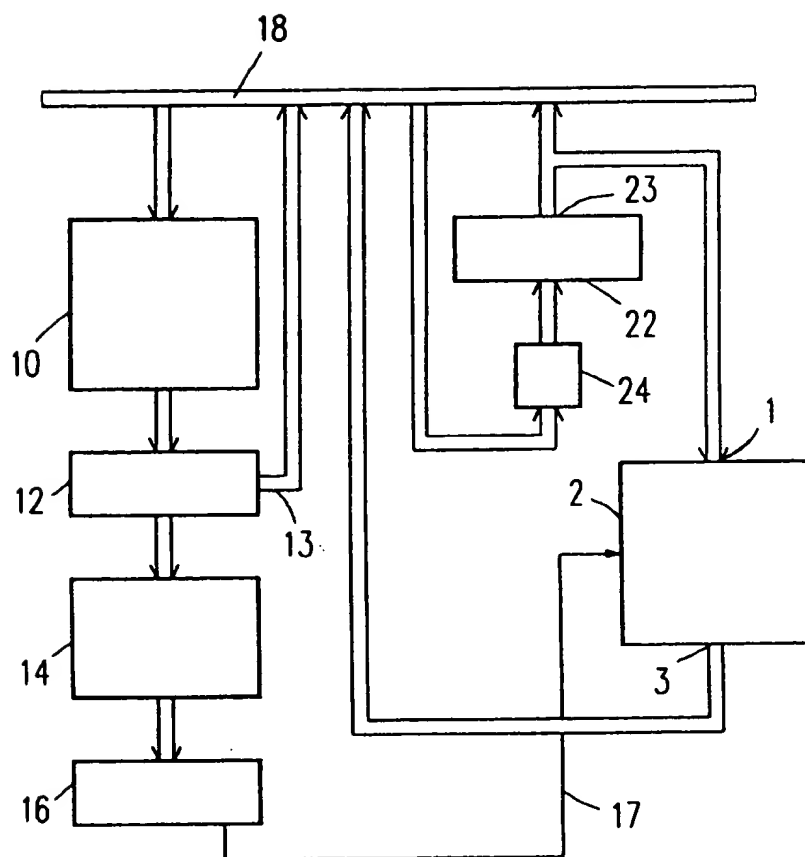


Fig.1

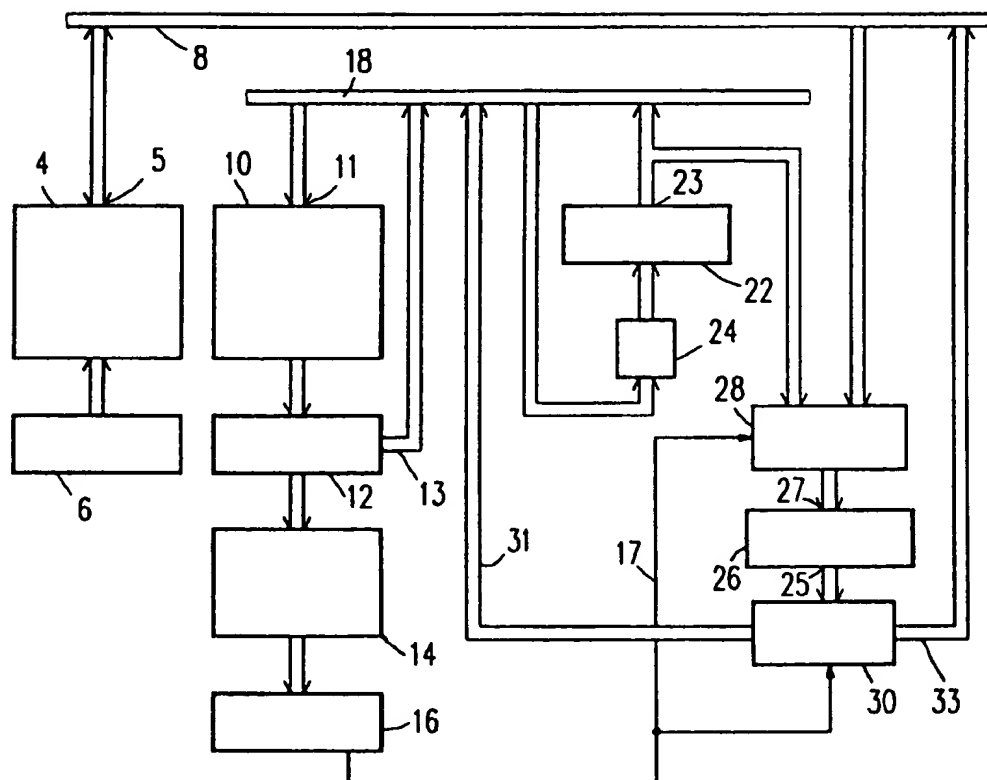


Fig.2



Europäisches
Patentamt

EUROPÄISCHER RECHERCHENBERICHT

Nummer der Anmeldung

EP 93 20 1005

EINSCHLÄGIGE DOKUMENTE			
Kategorie	Kennzeichnung des Dokuments mit Angabe, soweit erforderlich, der maßgeblichen Teile	Betrifft Anspruch	KLASSIFIKATION DER ANMELDUNG (Int. Cl.5)
X	WO-A-8 201 429 (NCR CORPORATION) * Zusammenfassung *	1	G06F9/42
X	IBM TECHNICAL DISCLOSURE BULLETIN Bd. 22, Nr. 8B, Januar 1980, NEW YORK US Seiten 3775 - 3777 J.A.KISELAK ET AL. 'Microprogram Link/Return Mechanism' * Abbildung 1 * * Seite 3776, Zeile 16 - Zeile 20 *	1	
P,X	PATENT ABSTRACTS OF JAPAN vol. 16, no. 457 (P-1426)22. September 1992 & JP-A-41 62 135 (NEC) 5. Juni 1992 * Zusammenfassung *	1	
A	FR-A-2 269 145 (NIXDORF COMPUTER AG) * Abbildung 1 * * Seite 5, Zeile 23 - Zeile 31 *	1	
			RECHERCHIERTE SACHGEBIETE (Int. Cl.5)
			G06F
Der vorliegende Recherchenbericht wurde für alle Patentansprüche erstellt			
Recherchenamt DEN HAAG		Abschließdatum der Recherche 26 JULI 1993	Prüfer SCHARFENBERGER B.
KATEGORIE DER GENANNTEN DOKUMENTE			
X : von besonderer Bedeutung als als betrachtet Y : von besonderer Bedeutung in Verbindung mit einer anderen Veröffentlichung derselben Kategorie A : technologischer Hintergrund O : schriftliche Offenbarung P : Zwischenliteratur		T : der Erfindung zugrunde liegende Theorien oder Grundsätze E : älteres Patentdokument, das jedoch erst am oder nach dem Anmeldedatum veröffentlicht worden ist D : in der Anmeldung angeführtes Dokument L : aus anderen Gründen angeführtes Dokument ----- A : Mitglied der gleichen Patentfamilie, überbrückendes Dokument	